Yin et al. EURASIP Journal on Wireless Communications and Networking (2016) 2016:210
DOI 10.1186/s13638-016-0710-5

EURASIP Journal on Wireless Communications and Networking

RESEARCH Open Access

A reinforced collaborative filtering approach based on similarity propagation and score predication graph

Xiaofei Yin[1,2], Tianye Chen[1,2], Wenrui Liu[3], Rong Xiao[1,2*], Chuanxiang Ma[1,2] and Zhongwang Fu[1,2]

Abstract

In the era of big data, the rapid development of mobile participatory sensing devices brings the explosive expansion of data, making information overload a serious problem. In this case, a personalized recommendation system on mobile social media appears. Collaborative filtering is the most widely used approach in a recommendation system. Nevertheless, there still exist many problems, such as the serious data sparsity problem and the cold start problem. Existing approaches cannot effectively solve these problems. Most of the existing recommendation approaches are based on single information source and cannot effectively solve the cold start and data sparsity problems. In addition, some approaches proposed to solve data sparsity fail to consider the effects of users' influences and prediction order on recommendation accuracy. Accordingly, from the perspective of increasing the categories of information, the similarity propagation approach based on a heterogeneous network is proposed to ease the cold start problems by improving the similarity calculation method. In addition, to ease the data sparsity problems, we propose a hybrid collaborative filtering approach based on a score prediction graph to finish the user-item score matrix in order. Finally, we conduct validation experiments on the MovieLens dataset. Compared with five state-of-the-art approaches, our approach outperforms them in terms of the performances of mean absolute error, root-mean-square error, recall, and diversity.

Keywords: Big data, Recommendation system, Similarity propagation, Heterogeneous network, Score prediction graph

1 Introduction

With the rapid development of mobile Internet, the mobile social media services [1, 2] are increasingly abundant. The mobile phone as a representative of mobile participatory sensing devices has become a part of people's daily life. However, the limited capacity of mobile users to receive information and the explosive growth of information in the mobile environment makes it difficult for mobile users to choose what they need from a lot of information quickly and effectively in the era of big data. Recommendation system plays an indispensible role in solving this problem.

Collaborative filtering is one of the most widely used approaches in a recommendation system. Nevertheless, there still exist many problems, such as the serious data sparsity problem and the cold start problem. In order to solve the two problems, many researchers have put forward several solutions. These studies can be divided into two categories. The first category is filling the user-item matrix by default or by prediction [3]. Filling the user-item matrix by default is inefficient and erases users' personalized information. Filling the user-item matrix by prediction is to predict scores according to the nearest neighbors of users or items. It is usually a one-time effort without considering the effects of users and user prediction order on recommendation accuracy, resulting in predication deviation on its nearest neighbors.

The other category is improving users' interest model and focusing on a certain aspect of users' or items' information to reduce the data sparseness. For example, the

* Correspondence: 20040363@hubu.edu.cn
[1]School of Computer Science and Information Engineering, Hubei University, Wuhan, Hubei 430062, China
[2]Educational Informationalization Engineering Research Center of Hubei Province, Wuhan, Hubei 430062, China
Full list of author information is available at the end of the article

item-based collaborative filtering approach focuses on the scores of items in the nearest neighbor selection and the content-based collaborative filtering approach focuses on the content of items to build users' interest model [4]. This kind of solution, relying on a single information source, fails to satisfy users' diversified demand due to its inaccuracy and single recommendation.

Accordingly, in this paper, we first propose a similarity propagation approach based on a heterogeneous network to effectively ease the cold start and data sparsity problems. The proposed similarity propagation approach based on heterogeneous networks analyzes users' preferences from multi-perspectives by combining several types of information, which overcomes the drawbacks and disadvantages caused by single information source. Then we propose a score prediction graph (SPGraph) generation approach and work out a prediction node sequence under the principle that the less influence a node has, the earlier it will be predicted. Based on the prediction node sequence, we fill the user-item matrix step by step to generate a recommendation list, which can reduce the impact of a node's predication deviation on its nearest neighbors to really improve the prediction accuracy.

The main contributions of this paper are summarized as follows:

1. We integrate several types of information and relations into recommendation heterogeneous networks and propose the similarity propagation approach which mitigates the impacts of cold start and data sparsity problems caused by single information source.
2. We propose the prediction node sequence generation approach-based SPGraph to improve accuracy by reducing the impact of a node's predication deviation on its nearest neighbors.
3. We conduct sufficient experiments on the MovieLens dataset, which demonstrate that our approach outperforms five state-of-the-art approaches.

The rest of this paper is organized as follows: Section 2 summarizes the related work. Section 3 thoroughly demonstrates the proposed similarity propagation approach. Section 4 explicates the details of the SPGraph-based collaborative filtering approach. Section 5 shows the experimental data and results along with a thorough analysis. Lastly, Section 6 concludes the paper and discusses the future work.

2 Related work

In this section, we briefly review the existing recommendation approaches which fall into four main categories: collaborative filtering recommendation, content-based recommendation, knowledge-based recommendation, and hybrid recommendation.

2.1 Collaborative filtering recommendation

It serves to predict and recommend the items that target users might like according to the interests of their nearest neighbors who share with them the similar behavioral characteristics obtained from the analysis of their behavioral habits [5]. Recent years have witnessed an endless stream of studies and researches on collaborative filtering approaches which can be divided into two categories: neighborhood-based and model-based. Neighborhood-based approaches, further divided into user-based [6] and item-based approaches [7, 8], serve to identify similar users with the target user according to their similarity which is measured by their feedbacks on shared items and then compute predictions based on these similar users' feedbacks on other items. It is faced with the feedback scarcity that arises in practice because a user may only give feedbacks on a limited number of items, namely data sparsity. Model-based approaches, such as aspect models [9], latent factor models [10], Bayesian models [11], and decision trees, alleviate the feedback scarcity by generating a global model based on the given training data and then using the model to predict the active user's preference on unknown items, but most of them suffer from high computational overheads caused by the tuning of a large number of parameters embedded in the models. As a result, it is hard to apply them into large-scale social networks.

2.2 Content-based recommendation

It is realized by matching users' characteristics with items' content, which has been studied in many papers. For example, Yu et al. [12] proposed the recommendation approach for multiple interests and multiple contents. Hannon et al. [13] put forward the UPR model used for twitter forward recommendation. Wu et al. [14] combined content-based recommendation with system-based recommendation to predict and recommend according to the CCAM model. Ronen et al. [15] studied a content-based characteristic selection method which is independent of recommendation systems. Most existing content-based recommendation systems, in which items are usually described with keywords, are designed to recommend items according to text contents. However, similarity evaluations based on keywords may be misleading due to the ambiguity of natural languages. Besides, this kind of approaches may also result in deviation of the results, of which the single and inaccurate information source is the root cause.

2.3 Knowledge-based recommendation

This kind of recommendation is closely linked and sometimes even interactive with users' requirements. When

users input their requirements, the system will work out recommendation results to match. If no results show up, users will have to modify their requirements. Burke [16] proposed the restraint-based recommendation system based on recommendation knowledge base while in [17] Burke proposed the case-based recommendation approach.

2.4 Hybrid recommendation

Hybrid recommendation systems are the integrative, parallel, or linear combinations of several recommendation systems with an effort to fill in the gaps of single recommendation systems. Top-N based collaborative filtering (TNCF) and majorizing similarity based collaborative filtering (MSCF) [18] proposed by Song are hybrid collaborative filtering approaches which integrate score similarity and property similarity. They first compute user similarity and select the top N nearest neighbors of the target user and then predict scores and provide recommendation. This method improves the accuracy, while it greatly increases the complexity of the computation.

Collaborative filtering recommendation, content-based recommendation, and knowledge-based recommendation approaches are all based on a single information source and fail to satisfy users' diversified demand and effectively solve the cold start and data sparsity problems.

Although hybrid recommendation approaches try to overcome the cold start and data sparsity problems by combing several recommendation systems, they are just linear combinations and cause high approach complexity and non-accurate prediction.

In addition, these approaches proposed to solve data sparsity fail to consider the effects of users' influences and prediction order on recommendation accuracy.

3 Similarity propagation approach based on heterogeneous networks

In this section, we propose a similarity propagation approach based on heterogeneous networks to overcome the cold start and data sparsity problems. We first define some terms used in our paper. Then we describe our similarity propagation approach based on heterogeneous networks.

3.1 Preliminaries

3.1.0.1 Definition 1 *Recommendation heterogeneous network*: As shown in Fig. 1, a recommendation heterogeneous network is made up of four major entities, namely users, items, tags, and properties. Six types of such entity relations mainly exist on the network, as *UP* (between users and properties), *UI* (relations between users and items), *UT* (relations between users and tags), *IP* (relations between items and properties), *IT* (relations between items and tags), and H (relations between homogeneous entities).

A recommendation heterogeneous network can be represented as $G_r = (V,E,W)$, where $V = V_u \cup V_i \cup V_t \cup V_p$, $E = E_{UP} \cup E_{UI} \cup E_{UT} \cup E_{IP} \cup E_{IT} \cup E_H$, and W is the weight of the relations. V is the union set of V_u, V_i, V_t, and V_p; V_u is the user set; V_i is the item set; V_t is the tag set; and V_p is the property set. E is the union set of E_{UP}, E_{UI}, E_{UT}, E_{IP}, E_{IT}, and E_H; E_{UP} is the relation between the user and the property; E_{UI} is the relation between the user and the item; E_{UT} is the relation between the user and the tag; E_{IP} is the relation between the item and the property; E_{IT} is the relation between the item and the tag; and E_H is the relation between homogeneous entities.

We define the following rules to determine whether relations exist between entities.

- If a user u possesses the property p, then $<u,p> \in E_{UP} \in E$
- If a user u purchases the item i and grades it as d, then $<u,i> \in E_{UI} \in E$ when and only when $d > \overline{d}$.
- If a user u is tagged as t, then $<u,t> \in E_{UT} \in E$.
- If an item i possesses the property p, then $<i,p> \in E_{IP} \in E$.
- If an item i is tagged as t, then $<i,t> \in E_{IT} \in E$.

3.1.0.2 Definition 2 *Meta path*: A meta path is defined as the path whose length is between two random nodes v_i and v_j in the recommendation heterogeneous network, denoted as $v_i \rightarrow^{l_{im}} v_m \rightarrow^{l_{mj}} v_j$, where l_{im} and l_{mj} represent certain types of relations, either of the same type or of different types.

There are three types of meta paths between users in the recommendation heterogeneous network, as shown in Fig. 2.

3.1.0.3 Property 1 Meta paths represent the similarities between entities.

Users purchasing the same item, users labeled with the same tag, and users possessing the same property all share some similarities. The more items, tags, and properties they share, the more similar they are.

3.1.0.4 Property 2 Similarities can transit to entities with no meta paths as long as they are connected with another common entity by at least one meta path.

For example, in Fig. 3, although there is no meta path between u_1 and u_3, they still share some similarities because both u_1 and u_3 are connected with u_2 by meta paths. Between u_1 and u_3 exists a random walking path composed of two or more meta paths which in nature are special random walking paths composed of only one meta path.

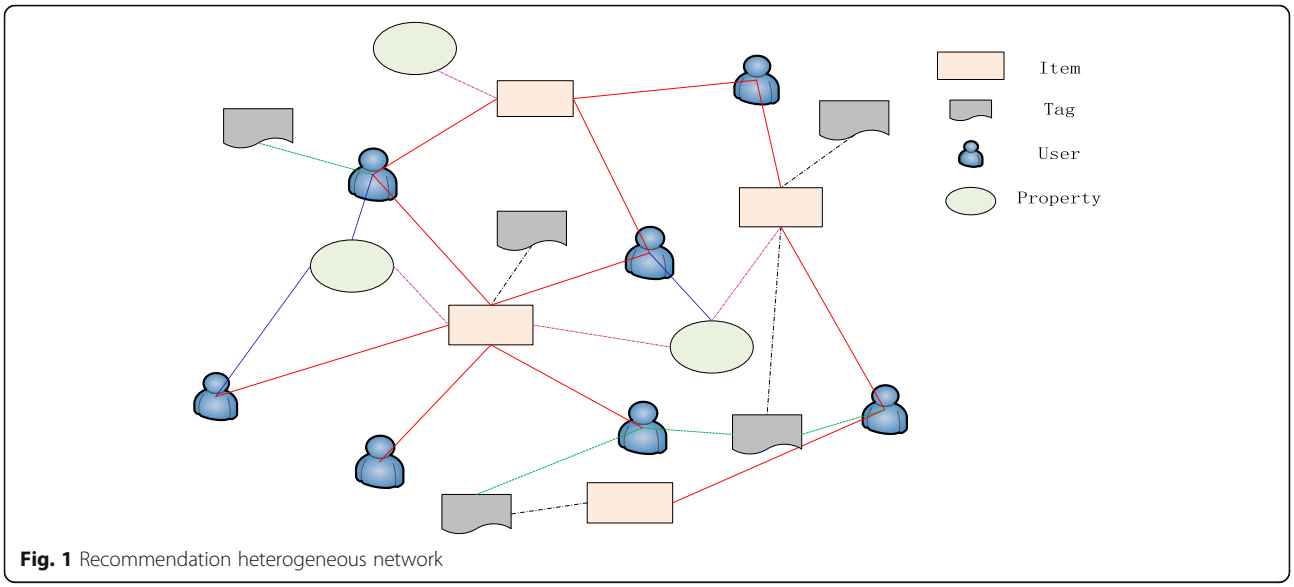

Fig. 1 Recommendation heterogeneous network

3.1.0.5 Definition 3 *Similarity propagation matrix*: Similarity propagation matrix can be defined as follows:

$$T = \begin{bmatrix} T_{UU} T_{UI} T_{UT} T_{UP} \\ T_{IU} T_{II} T_{IT} T_{IP} \\ T_{TU} T_{TI} T_{TT} T_{TP} \\ T_{PU} T_{PI} T_{PT} T_{PP} \end{bmatrix} \quad (1)$$

where U is users set, I is items set, T is tags set and P is properties set.

Similarity propagation matrix belongs to symmetric matrix. $t_{uv} \in T_{UU}$ is the similarity propagation probability between user u and user v. $t_{ui} \in T_{UI}$ is the similarity propagation probability between user u and item i. $t_{ut} \in T_{UT}$ is the similarity propagation probability between user u and tag t. $t_{up} \in T_{UP}$ is the similarity propagation probability between user u and property p. $t_{ij} \in T_{II}$ is the similarity propagation probability between item i and item j. $t_{it} \in T_{IT}$ is the similarity propagation probability between item i and tag t. $t_{ip} \in T_{IP}$ is the similarity propagation probability between item i and property p. $t_{mn} \in T_{TT}$ is the similarity propagation probability between tag m and tag n. $t_{tp} \in T_{TP}$ is the similarity propagation probability between tag t and property p. And $t_{pq} \in T_{PP}$ is the similarity propagation probability between property p and property q.

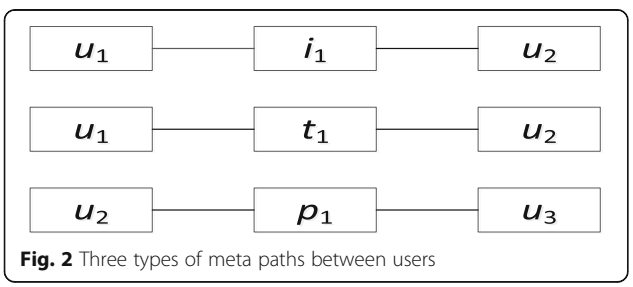

Fig. 2 Three types of meta paths between users

During the process of random walking, different types of relation accounts for various degrees of contribution and therefore are given different weights—w_{up}, w_{ui},w_{ut},-w_{ip}, and w_{it} for E_{UP}, E_{UI}, E_{UT}, E_{IP}, and E_{IT}, respectively. And the weight of relations between homogeneous entities is set as β. These parameters are defaulted as 1 in the experiments of this paper. The initialization of each sub-matrix in T is as follows.

3.1.1 Initialization of user probability propagation matrix

T_{UU} is the user probability propagation matrix, and the similarities between users are set as the number of the initial propagation matrixes. When users grade the same item, the improved Pearson coefficient will be used to measure the similarities between them.

The similarity between user u_i and user u_j is defined according to Eq. (2):

$$\mathrm{sim}(u_i, u_j) = \frac{\sum_{p \in P} \left(r_{u_i,p} - \overline{r_{u_i}}\right)\left(r_{u_j,p} - \overline{r_{u_j}}\right)}{\sqrt{\sum_{p \in P} \left(r_{u_i,p} - \overline{r_{u_i}}\right)^2} \sqrt{\sum_{p \in P} \left(r_{u_j,p} - r_{u_j}\right)^2}} \quad (2)$$

where P is the common items that u_i and u_j have graded and $\overline{r}_{u_i}$ and $\overline{r}_{u_j}$ are the average score of u_i and u_j, respectively.

If the users have no common grading items, sim(u_i,u_j) will be defined according to Eq. (3):

$$\mathrm{sim}(u_i, u_j) = \frac{P_i \cap P_j}{P_i \cup P_j} \quad (3)$$

where p_i is the items user u_i has purchased and p_j is the items user u_j has purchased.

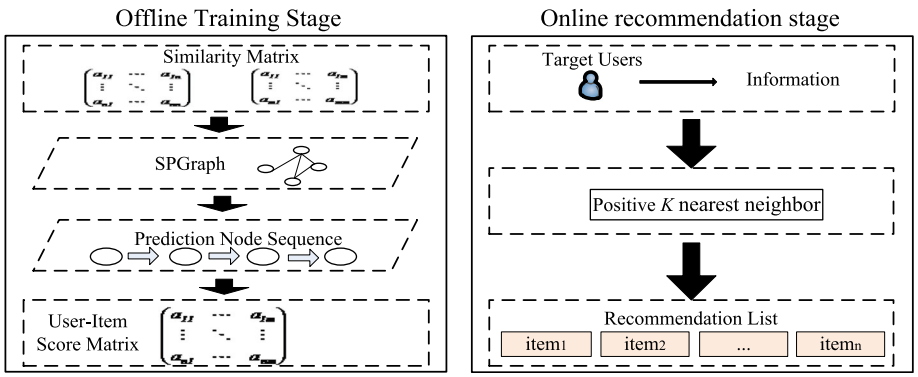

Fig. 3 Frame of SPGraph-based collaborative filtering approach

The formula of the sub-matrix of T_{UU} is as follows:

$$T_{UU}(i,j) = \begin{cases} \beta \mathrm{sim}(u_i, u_j) & \text{if } P \neq \varnothing \\ \dfrac{P_i \cap P_j}{P_i \cup P_j} & \text{otherwise} \end{cases} \quad (4)$$

where β is the weight of the relation between homogeneous entities.

3.1.2 Initialization of user-item probability propagation matrix

T_{UI}, denoting the user-item probability propagation matrix, is defined as follows:

$$T_{UI}(i,j) = \begin{cases} w_{ui} \cdot 1 & \text{if } e_{ui} = 1 \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

where w_{ui} is the weight of the relation between users and items.

When user u_i purchased item i_j and graded it as s, if s is larger than the threshold value δ, $e_{ui} = 1$; otherwise, $e_{ui} = 0$.

3.1.3 Initialization of user-tag probability propagation matrix

T_{UT} is the user-tag probability propagation matrix. We employ the term frequency–inverse document frequency (TF-IDF) approach to measure the similarity between users and tags. The more often user u_i uses or is labeled with tag t_j and the less popular tag t_j is, the more similar user u_i and tag t_j are.

T_{UT} is defined as follows:

$$T_{UT}(i,j) = \begin{cases} w_{ut} \dfrac{n_{u,t}}{\log(1 + n_{t^{(u)}})} & \text{if } e_{ut} = 1 \\ 0 & \text{if } e_{ut} = 0 \end{cases} \quad (6)$$

where $n_{u,t}$ is the times user u_i uses tag t_j, $n_t^{(u)}$ is the times tag t_j is used, and w_{ut} is the weight of the relation between users and tags. $e_{ut} = 1$ indicates that the user has used the tag while $e_{ut} = 0$ indicates the user has not used the tag.

3.1.4 Initialization of user-property probability propagation matrix

T_{UP}, denoting the user-property probability propagation matrix, is defined as follows:

$$T_{UP}(i,j) = \begin{cases} w_{up} \cdot 1 & \text{if } e_{up} = 1 \\ 0 & \text{otherwise} \end{cases} \quad (7)$$

where w_{up} is the weight of the relation between users and properties. If user u_i possesses property p_j, then $e_{up} = 1$; otherwise, $e_{up} = 0$.

3.1.5 Initialization of item probability propagation matrix

$T_{II}$is the item probability propagation matrix, and the similarities between items are set as the number of the initial propagation matrixes. When item I_i and item I_j are graded by a common user, the improved Pearson coefficient will be used to measure the similarities between them.

The similarities between two item entities, denoted as $\mathrm{sim}(I_i,I_j)$, is defined according to Eq. (8):

$$\mathrm{sim}(I_i, I_j) = \frac{\sum_{u \in U} \left(r_{u,I_i} - \bar{r}_{u_i}\right)\left(r_{u,I_j} - \bar{r}_{u_j}\right)}{\sqrt{\sum_{u \in U} \left(r_{u_i,p} - \bar{r}_{u_i}\right)^2} \sqrt{\sum_{p \in P} \left(r_{u_j,p} - r_{u_j}\right)^2}} \quad (8)$$

where U is the users who have graded both I_i and I_j and $\bar{r}_{u_i}$ and $\bar{r}_{u_j}$ are the average grades of u_i and u_j, respectively.

If I_i and I_j have not been graded by a common user, $\mathrm{sim}(I_i,I_j)$ will be defined according to Eq. (9):

$$\mathrm{sim}(I_i, I_j) = \frac{U_i \cap U_j}{U_i \cup U_j} \quad (9)$$

where U_i is the users who have purchased item I_i while U_j is the users who have purchased item I_j.

The formula of the sub-matrix of T_{II} is as follows:

$$T_{II}(i,j) = \begin{cases} \beta \mathrm{sim}(I_i, I_j) & \text{if } P \neq \varnothing \\ \dfrac{U_i \cap U_j}{U_i \cup U_j} & \text{otherwise} \end{cases} \quad (10)$$

where β is the weight of the relation between homogeneous entities.

3.1.6 Initialization of item-tag probability propagation matrix

T_{IT} is the user-property probability propagation matrix. We employ the TF-IDF approach to measure the similarity between items and tags. The more often item I_i is labeled with tag T_j and the less popular tag T_j is, the more similar item I_i and tag T_j are.

T_{IT} is defined as follows:

$$T_{IT} = \begin{cases} w_{it} \dfrac{n_{i,t}}{\log(1+n_t^{(i)})} & \text{if } e_{it} = 1 \\ 0 & \text{otherwise} \end{cases} \quad (11)$$

where $n_{i,t}$ is the times item I_i is labeled with tag T_j, $n_t^{(i)}$ is the times tag T_j is used, and w_{it} is the weight of the relation between items and tags. $e_{it} = 1$ indicates that the item has been labeled with the tag.

3.1.7 Initialization of item-property probability propagation matrix

T_{IP}, denoting the item-property probability propagation matrix, is defined as follows:

$$T_{IP}(i,j) = \begin{cases} w_{ip} \cdot 1 & \text{if } e_{ip} = 1 \\ 0 & \text{otherwise} \end{cases} \quad (12)$$

where w_{ip} is the weight of the relation between items and properties. If item I_i possesses property p_j, then $e_{ip} = 1$; otherwise, $e_{ip} = 0$.

3.1.8 Initialization of tag probability propagation matrix

T_{TT}, denoting the tag probability propagation matrix, refers to the similarities between tags and is defined as follows:

$$T_{TT}(i,j) = \begin{cases} w_{tt} \dfrac{\sum_{i \in N(b) \cap N(b')} n_{b,i} n_{b',i}}{\sqrt{\sum_{i \in N(b)} n_{b,i}^2 \sum_{i \in N(b')} n_{b',i}^2}} \\ 0 \end{cases} \quad (13)$$

where w_{tt} is the weight of the relation between tags, $N(b)$ is the tag set containing tag b, and $n_{b,j}$ is the number of users that label item i with tag b.

3.1.9 Initialization of property-tag probability propagation matrix

T_{TP}, denoting the property-tag probability propagation matrix, is a null matrix as a result of the lack of direct relations between properties and tags.

3.1.10 Initialization of property probability propagation matrix

T_{PP}, denoting the property probability propagation matrix, is also a null matrix because no direct relations exist between properties.

Finally, we normalize to 1 each line of the propagation probability matrix.

3.2 Similarity propagation approach

If a d-length path exists between two random nodes v_o and v_s in a heterogeneous network, it will take d random walks from v_o to v_s and the path between v_o and v_s will be a d-length random path. When arriving at node v_t during the random walking, we can either proceed to another node at the propagation probability between node v_t and its neighbor node or restart at the certain probability α [19]. Until the probability of access to each node converges to a number, it ceases to propagate. Both methods will lead to a Markov chain.

Random walking paths are made up of meta paths which represent the similarities between entities. And the similarities, propagated during the process of random walking, are positively correlated with the number of random walking paths and negatively correlated with the length of them.

Therefore, the formula of the similarity propagation between v_i and v_j is defined according to Eq. (14):

$$\mathrm{sim}(v_i, v_j) = \sum\nolimits_{\gamma \in l} p(\sigma) \alpha (1-\alpha)^{\mathrm{length}(\gamma)} \quad (14)$$

where l is a path from v_i to v_j and length(γ) is a γ-length path from v_i to v_j.

Turning the above formula into a matrix, we get a similarity matrix, defined as follows:

$$R_{\mathrm{sim}} = \begin{bmatrix} R_{UU} & R_{UI} & R_{UT} & R_{UP} \\ R_{IU} & R_{II} & R_{IT} & R_{IP} \\ R_{TU} & R_{TI} & R_{TT} & R_{TP} \\ R_{PU} & R_{PI} & R_{PT} & R_{PP} \end{bmatrix} = \sum_{\gamma=1}^{l} \alpha (1-\alpha)^{\gamma} T^{\gamma} \quad (15)$$

where R_{UU} is the similarity matrix of users and R_{II} is the similarity matrix of items.

4 SPGraph-based collaborative filtering approach

Based on the user similarity matrix and item similarity matrix deduced from the similarity propagation approach, this section proposes a hybrid collaborative filtering approach based on the score prediction graph (SPGraph). Figure 3 illustrates the framework of our approach, which consists of two stages.

The offline training stage constructs the SPGraph by searching for the nearest neighbors of users or items via

similarity matrix, generates prediction node sequence by anti-centrality sort principle after calculating the centrality of each node, and finishes the user-item score matrix via the hybrid collaborative filtering approach. The online recommendation stage searches for the positive K nearest neighbors of the target user via the similarity propagation approach and predicts users' scores via the hybrid collaborative filtering approach to form recommendation lists.

4.1 Construction of SPGragh

4.1.0.1 Definition 4 *SPGraph*: SPGraph is an isomorphic undigraph with weight generated by the nearest neighbor selection in the user similarity matrix and the item similarity matrix. A SPGraph can be represented as SPGraph = (V,E,W), where V is one type of entity, either users or items, E denotes one type of relation, and W is the similarity between entities. $<v_i,v_j>\in E$ means that item v_i and item v_j or user v_i and user v_j have similarity which is represented as $w_{ij}\in W$.

Then we will introduce the SPGraph generation approach with the example of the user similarity matrix. And the user score predication graph is denoted as SPG_U. Near neighbors with low similarity not only occupy computing resources but also reduce the accuracy of predication; therefore, in the nearest neighbor selection, we set a threshold value δ to eliminate them. Approach 1 presents the pseudo-code of SPGraph generation approach.

Approach 1.SPGraph generation approach

Input：user similarity matrix R_{UU}, SPGraph's adjacent matrix $SPG_U[N*N]$, the number of users N, threshold value δ。
Output: $SPG_U[N*N]$

```
1. begin
2.for    i=1    to    N
3.   for    j=i    to    N
4.      SPG_U(i,j) = SPG_U(j,i) =0
5.   end for
6. end for
7. for    i=1    to    N
8.    for    j=i+1    to    N
9.      If ( r_ij > δ )    SPG_U(i,j) = SPG_U(j,i) = r_ij
10.    end If
11.   end for
12. end for
13. return SPG_U
14. end
```

At the beginning, there only exists the set of independent user nodes $V = \{v_1,v_2,\ldots,v_n\}$ in SPG_U. Lines 7–12 demonstrate that, for every value in the matrix ($r_{ij}\in R_{UU}$), if r_{ij} is greater than δ, an edge $<v_i,v_j>$ will be added to the SPG_U and the weight of the edge w_{ij} is set to r_{ij}. Figure 4(1) briefly shows the generation approach of SPG_U.

4.2 Generation of prediction node sequence based on anti-centrality sort

As is shown in Fig. 4(2), the centrality of user u_5 is the lowest. So if we first predicate the score of u_5, its deviation will only affect the score predication of u_6 to some extent and have little effect on other users. However, if we first predicate the score of u_3, its deviation will directly affect the score prediction of u_1, u_4, and u_6. The score prediction of u_4 will be affected the most due to its high similarity with u_3. According to the principle "The less influential the node is, the lower centrality the node has, the earlier the node is predicted, the less rating error is.", we propose a prediction node sequence generation approach based on anti-centrality sort. The pseudo of the prediction node sequence generation approach is as follows.

Approach 2. prediction node sequence generation approach based on anti-centrality sort（taking users as the example）

Input: users score predication graph SPG_U
Output: set of user sequence userSquence

```
1. begin
2. //N is the number of users, Sum stores the centrality of nodes
3. userSquence [N],Sum[N];
4. for   i=1    to    N
5.    for    j=1    to    N
6.      Sum[i]=Sum[i]+SPG_U[i][j];
7.    end for
8. end for
9. for   m=1    to    N
10. //find the node with the lowest centrality
11.   q=FindMin(Sum);
12.   userSquence [m]=q;
13. //delete the node added to the userSquence and its similarity
14.   for   n=1    to    N
15.      Sum[n]=Sum[n]-SPG_U[n][q];
16.   end for
17. end for
18. return userSquence
19. end
```

Lines 4–8 describe the computing of the centrality of nodes in SPG_U. Lines 11–12 describe that we first search for the node with the lowest centrality and then add it to the node sequence array in order. Lines 14–16 describe that we delete the node added to the node sequence array and its similarity, re-compute the centrality of rest, and then repeat the process from line 10 to line 16 till the completion of the node sequence. The generating process of the prediction node sequence of SPG_U is as shown in Fig. 4(2).

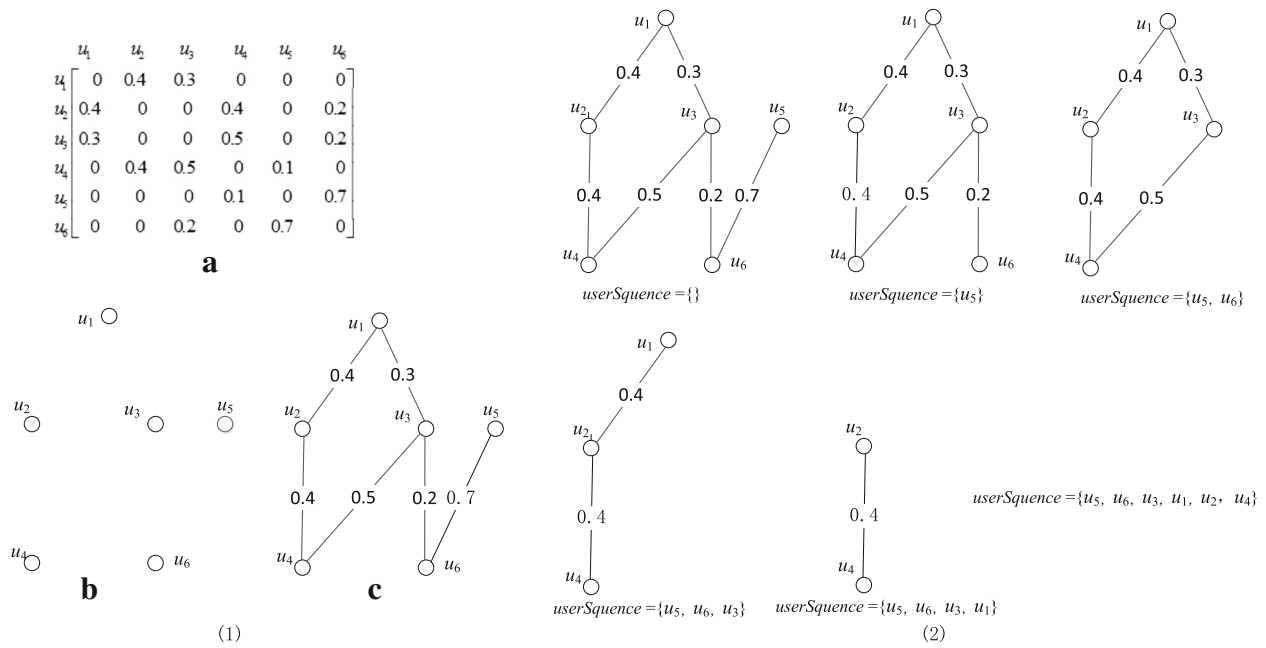

Fig. 4 User predication node sequence generation approach

4.3 Hybrid collaborative filtering approach

Based on the prediction node sequence, we use the hybrid approach integrating the user-based nearest neighbor recommendation approach (UbCF) with the item-based nearest neighbor recommendation approach (IbCF) to predict the score that user u gives to item i.

Based on the UbCF approach, the predicated score user u gives to item i is computed according to Eq. (16).

$$\mathrm{pred}(u,i) = \bar{r_u} + \frac{\sum_{s\in \mathrm{Sim}(u)} \mathrm{sim}(u,s) * \left(r_{s,i} - \bar{r_s}\right)}{\sum_{s\in \mathrm{Sim}(i)} \mathrm{sim}(u,s)} \quad (16)$$

where Sim(u) is the nearest neighbors of user u, sim(u,s) is the similarity between user u and user s, $r_{s,i}$ is the score that user s gives to item i, and $\bar{r}_s$ is the average score that user s gives to all items.

Based on the IbCF approach, the predicated score user u gives to item i is computed according to Eq. (17):

$$\mathrm{pred}(u,i) = \frac{\sum_{p\in \mathrm{Sim}(i)} \mathrm{sim}(i,p) * r_{u,p}}{\sum_{p\in \mathrm{Sim}(i)} \mathrm{sim}(i,p)} \quad (17)$$

where Sim(i) is the nearest neighbors of item i, sim(i,p) is the similarity between item i and item p, and $r_{u,p}$ is the score user u gives to item p.

Due to the impact of the similarity of the near neighbor set, UbCF and IbCF vary in the accuracy of predication. For example, it is obvious that the UbCF-based recommendation results are more accurate when the similarity of the user's near neighbor set is {1,0.8,0.9} while the similarity of the item's near neighbor set is {0.4,0.5,0.5}. So the confidence weight [20] is introduced to balance the final prediction result. And the larger the similarity of the near neighbors set is, the bigger its confidence weight is.

The confidence weight of the user is defined according to Eq. (18):

$$\mathrm{con}_u = \sum_{u_i\in \mathrm{Sim}(u)} \frac{\mathrm{sim}(u_i,u)}{\sum_{u_i\in \mathrm{Sim}(u)} \mathrm{sim}(u_i,u)} \times \mathrm{sim}(u_i,u) \quad (18)$$

The confidence weight of the item is defined according to Eq. (19):

$$\mathrm{con}_v = \sum_{v_i\in \mathrm{Sim}(u)} \frac{\mathrm{sim}(v_i,v)}{\sum_{v_i\in \mathrm{Sim}(v)} \mathrm{sim}(v_i,v)} \times \mathrm{sim}(v_i,v) \quad (19)$$

Larger confidence weight results in more accurate predication.

Besides, different data sets and users may put varied weight on these two recommendation approaches. Therefore, parameter θ is introduced to measure the weight a user gives to an approach. w_u denoting the weight of the UbCF approach and w_v denoting the weight of the IbCF approach are defined as follows:

$$w_u = \frac{\mathrm{con}_u \times \theta}{\mathrm{con}_u \times \theta + \mathrm{con}_i \times (1-\theta)} \tag{20}$$

$$w_v = \frac{\mathrm{con}_i \times (1-\theta)}{\mathrm{con}_u \times \theta + \mathrm{con}_i \times (1-\theta)} \tag{21}$$

Furthermore, when neither the users' nearest neighbor set Sin(u) nor the items' nearest neighbor set Sin(i) are null sets, the hybrid recommendation approach is defined as follows:

$$\mathrm{pred}(u,i) = w_u \times \mathrm{pred}_u(u,i) + w_v \times \mathrm{pred}_v(u,i) \tag{22}$$

where the sum of w_u and w_v is 1.

$$w_u = \frac{\mathrm{con}_u \times \theta}{\mathrm{con}_u \times \theta + \mathrm{con}_i \times (1-\theta)} \tag{23}$$

$$w_v = \frac{\mathrm{con}_i \times (1-\theta)}{\mathrm{con}_u \times \theta + \mathrm{con}_i \times (1-\theta)} \tag{24}$$

When Sin(i) is a null set and Sin(u) is not, the hybrid recommendation approach equals to the UbCF. And when Sin(u) is a null set and Sin(i) is not, the hybrid recommendation approach equals to the IbCF.

If both Sin(u) and Sin(i) are null sets during the user-item matrix filling phase in the offline training phase, pred(u,i) = null. As to online prediction, cold start problems have been solved by similarity propagation approach because there will be a corresponding nearest neighbor set for every new user or item. Besides, in order to improve the accuracy of predication, we choose the positive K nearest neighbor instead of the top K nearest neighbor. And positive K is defined as follows:

$$\text{positive } K(u) = \{u_i | \mathrm{sim}(u_i, u) \geq \delta \ \} \tag{25}$$

The value of K varies in users as a result of the different number of users filtered by the similarity threshold value δ.

5 Experiments and comparison

In this section, we evaluate our approach. We first introduce the experiment dataset, the evaluation metrics, and the parameter setting. Then we perform some experiments to investigate the performance of our approach compared with five state-of-the-art approaches.

5.1 Datasets, evaluation metrics, and parameter setting

In this experiment, we employ the available movie datasets, MovieLens, which can be obtained from the MovieLens site [21]. Table 1 tabulates the details about the datasets.

The dataset is so sparse that we need to pre-process it by deleting users whose movie records are less than 50 and movies which are graded by less than 50 users.

Table 1 Statistics of experiment datasets

Statistics	MovieLens
Number of users	700
Number of movies	10,000
Number of tags	6100
Theme of movies	20
Rating records	100,000
Rating range	0–5

In the experiment, we employ four commonly used evaluation metrics including mean absolute error (MAE), root-mean-square error (RMSE), recall rate, and diversity. They are defined and summarized as follows.

$$\mathrm{MAE} = \frac{\sum_{u,i \in T} \left| r_{u,i} - r_{u,i} \right|}{|T|} \tag{26}$$

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{u,i \in T} \left(r_{u,i} - r_{u,i} \right)^2}{|T|}} \tag{27}$$

where $|T|$ is the number of rating records in the test set, $r_{u,i}$ is the actual score for user u on item i and $\overline{r}_{u,i}$ is the predicted score by recommendation system.

$$\mathrm{Recall} = \frac{\sum_{u \in U} |R(u) \cap P(u)|}{\sum_{u \in U} |P(u)|} \tag{28}$$

where $P(u)$ is the item set that target users graded in the test set and $R(u)$ is the item set recommended by the recommendation system.

$$\mathrm{Diversity}(R(i)) = 1 - \frac{\sum_{i,j \in R(u), i \neq j} \mathrm{sim}(i,j)}{\frac{1}{2}|R(u)|(|R(u)|-1)} \tag{29}$$

where sim(i,j) is the similarity between item i and item j and $R(u)$ is the recommendation list.

The diversity of the recommendation system is the mean value of the diversities of all the users' recommendation lists and is defined according to Eq. (30).

$$\mathrm{Diversity} = \frac{1}{|U|} \sum_{u \in U} \mathrm{Diversity}(R(u)) \tag{30}$$

where U is the user set in the test set.

The threshold values and parameters in our experiment are optimized with the restart factor 0.8, the threshold value 0.4, and the balance factor 0.9. The weights of the six types of relations are initialized as 1. For lack of space, the parameter optimization process is not mentioned in this paper. The threshold and the balance factor are optimized. Here only Fig. 5 is used to demonstrate the effects of the restart factor on the performance of the approach.

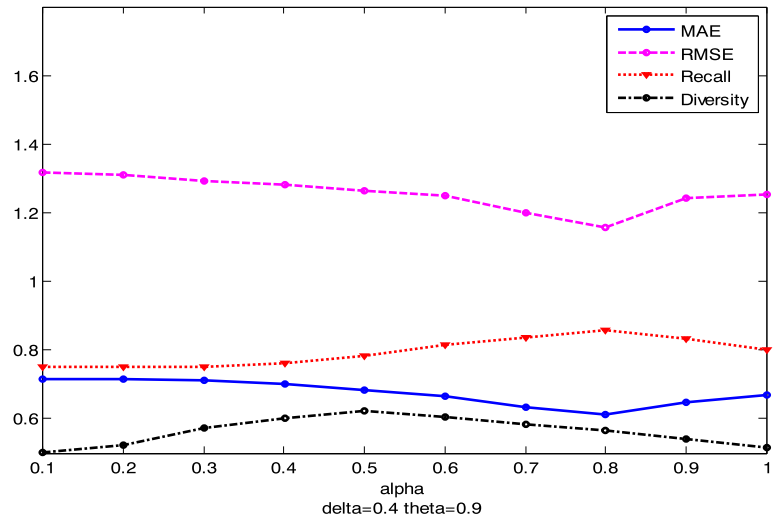

Fig. 5 The effect of restart factor alpha on the performance of the approach

5.2 Experiment results

In order to confirm whether our approach can perform better than other approaches, we compare our approach with five state-of-the-art approaches. More details are provided below:

UbCF and *IbCF* [22], whose information source is the scores users give to items, are user-based and item-based nearest neighbor recommendation approaches, respectively.

Hybrid user-item based collaborative filtering (*HCF*) [22] is the hybrid recommendation approach which integrates the user-based nearest neighbor recommendation with the item-based nearest neighbor recommendation.

TNCF is a hybrid recommendation approach which integrates the rating similarity with the property similarity. And it adopts the top N method to select the nearest neighbors.

MSCF is the reinforced approach of TNCF by improving the nearest neighbor selection method.

Then we randomly split the dataset into 10 subsets for 10-fold cross-validation. The nine subsets are training datasets, and the remaining one subset is the test dataset. All approaches will be repeated 20 times in every experiment to avoid sample bias. The mean values of the four evaluation metrics are calculated when the length of the recommendation list is 20 and 30. The comparison results are summarized in Table 2 and Fig. 5.

Figure 6 shows clearly the performance of the six approaches on MAE, recall, and diversity. It is very clear that the performance of HCF on all the four evaluation metrics is better than UbCF and IbCF because HCF is a hybrid recommendation approach which integrates the user-based nearest neighbor recommendation with the item-based nearest neighbor recommendation.

Table 2 Results of the six approaches

Methods	UbCF	IbCF	HCF	TNCF	MSCF	Our approach
	The length of recommendation list $n = 20$					
MAE	0.799000	0.820004	0.786254	0.759431	0.749642	0.612635
RMSE	1.220041	1.238451	1.190004	1.172457	1.162245	1.158036
Recall	0.752544	0.692545	0.792445	0.795474	0.798415	0.856674
Diversity	0.354954	0.352424	0.385454	0.401771	0.524854	0.568545
	The length of recommendation list $n = 30$					
MAE	0.795714	0.818571	0.776040	0.755981	0.746721	0.602412
RMSE	1.228245	1.245100	1.192544	1.175125	1.165105	1.142428
Recall	0.789254	0.725875	0.805747	0.814745	0.828604	0.855441
Diversity	0.354521	0.352488	0.385471	0.401545	0.524454	0.585145

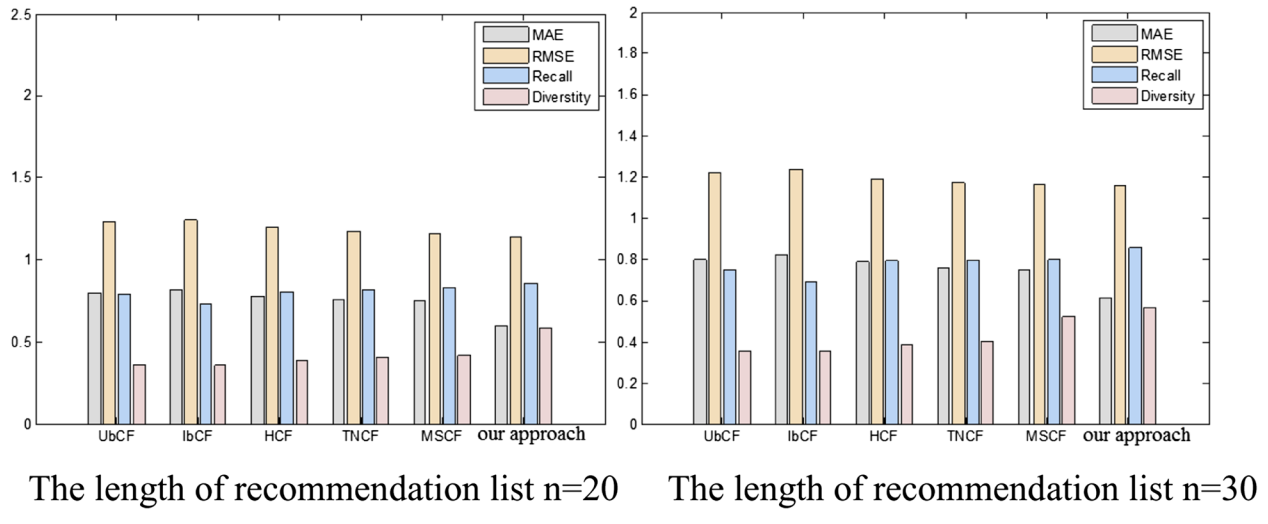

Fig. 6 Results of the six approaches

As for TNCF, its improvement on similarity computing by introducing property similarity alleviates the impact of sparse data to some extent. So TNCF performs better than HCF. On the other hand, MSCF performs better than TNCF as a result of its improvement on both similarity computing and nearest neighbor selection.

Among the six approaches, our approach outperforms other approaches in terms of the performance of MAE, RMSE, recall, and diversity. To be more specific, compared with MSCF, our approach improves 19.3 % in MAE, 1.9 % in RMSE, 7.2 % in recall rate, and 11.5 % in diversity. There are two reasons that our approach is the best. On the one hand, our approach adds other properties and information to similarity computing to further reduce the effect of sparse matrix. On the other hand, our approach uses the hybrid collaborative filtering approach to predicate scores and fill the user-item matrix step by step based on prediction node sequence which can really improve the prediction accuracy.

6 Conclusions

In this paper, we address the following issues. Firstly, most of the existing recommendation approaches are based on single information source and cannot effectively solve the cold start and data sparsity problems. In addition, some approaches proposed to solve data sparsity fail to consider the effects of users' influences and prediction order on recommendation accuracy. To solve these problems, the paper proposes the similarity propagation approach based on heterogeneous networks and the predication node sequence generation approach based on anti-centrality sort, the former integrating various types of information to effectively solve the cold start problem and the latter solving data sparsity by gradually filling the user-item score matrix based on prediction node sequence. We conduct experiments on the MovieLens dataset. Compared with five state-of-the-art approaches, our approach outperforms them in terms of the performances of MAE, RMSE, recall, and diversity.

There are several areas in which we can improve our work. Firstly, more feature extraction methods [23] can be introduced to analyze the user preference. Secondly, social community discovery [24] and precise semantic analysis method [25–28] can be introduced to the similarity computing so as to more accurately and effectively work out user preference. Thirdly, more methods [29] can be used to filter the training dataset in order to make the dataset trustworthy. Fourthly, we can implement the Spark-based similarity propagation approach to improve approach efficiency.

Acknowledgements

This work is partly supported by the grants of Science-technology Support Plan Projects of Hubei (2014BAA089) and Natural Science Foundation of Hubei (2011CDB072).

Competing interests

The authors declare that they have no competing interests.

Author details

[1]School of Computer Science and Information Engineering, Hubei University, Wuhan, Hubei 430062, China. [2]Educational Informationalization Engineering Research Center of Hubei Province, Wuhan, Hubei 430062, China. [3]State Key Laboratory of Software Engineering, School of Computer, Wuhan University, Wuhan, Hubei 430072, China.

Received: 28 May 2016 Accepted: 26 August 2016
Published online: 06 September 2016

References

1. JJPC Rodrigues, M Oliveira, B Vaidya, New trends on ubiquitous mobile multimedia applications. EURASIP J. Wireless. Commun. Netw. **2010**(1), 1 (2010)
2. L Yang, X Geng, H Liao, A web sentiment analysis method on fuzzy clustering for mobile social media users. EURASIP J. Wireless. Commun. Netw. **2016**(1), 1 (2016)

3. Breese J S, Heckerman D, Kadie C, Empirical analysis of predictive algorithms for collaborative filtering. Proceedings of the Conference on Uncertainty in Artificial Intelligence. (Madison, Wisconsin July 24-26, 1998), p. 43-52.
4. Yang W, Cui X, Liu J, et al., User's interests-based movie recommendation in heterogeneous network. International Conference on Identification, Information, and Knowledge in the Internet of Things (IIKI Beijing, China 2015). IEEE, 74-77
5. Y Jiang, J Liu, M Tang, X Liu, An effective web service recommendation method based on personalized collaborative filtering.Web Services (ICWS), 2011 IEEE International Conference on(Washington, DC, USA). IEEE, 211–218 (2011)
6. JL Herlocker, JA Konstan, A Borchers, J Riedl, An algorithmic framework for performing collaborative filtering, in *SIGIR '99: Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (ACM, New York, 1999), pp. 230–237
7. G Linden, B Smith, J York, Amazon.com recommendations: item-to-item collaborative filtering. IEEE Int. Comput. **7**(1), 76–80 (2003)
8. B Sarwar, G Karypis, J Konstan, J Reidl, Item-based collaborative filtering recommendation algorithms, in *WWW '01: Proceedings of the 10th International Conference on World Wide Web* (ACM, New York, 2001), pp. 285–295
9. T Hofmann, Latent semantic models for collaborative filtering. ACM. Trans. Inf. Syst. **22**(1), 89–115 (2004)
10. J Canny, Collaborative filtering with privacy via factor analysis, in *SIGIR '02: Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (ACM, New York, 2002), pp. 238–245
11. Y Zhang, J Koren, Efficient Bayesian hierarchical user modeling for recommendation system, in *SIGIR '07: Proceedings of the 30th Annual International ACM SIGIR Conference on Jb Research and Development in Information Retrieval* (ACM, New York, 2007), pp. 47–54
12. L Yu, L Liu, X Li, A hybrid collaborative filtering method for multiple-interests and multiple-content recommendation in E-Commerce. Expert. Syst. Appl. **28**(1), 67–77 (2005)
13. Hannon J, McCarthy K, Smyth B. 2011. Finding useful users on twitter: twittomender the followee recommender. Advances in information retrieval. Springer Berlin Heidelberg, 784-787.
14. ML Wu, CH Chang, RZ Liu, Integrating content-based filtering with collaborative filtering using co-clustering with augmented matrices. Expert. Syst. Appl. **41**(6), 2754–2761 (2014)
15. Ronen R, Koenigstein N, Ziklik E, et al, Selecting content-based features for collaborative filtering recommenders. ACM Conference on Recommender Systems, 407-410 2013
16. Burke, Knowledge-based recommender systems. In A. Kent (ed.), Vol. 69, Supplement 32. New York: Marcel Dekker, 180-200 2000.
17. Burke R, The Wasabi Personal Shopper: a case-based recommender system. Proceedings of the sixteenth national conference on artificial intelligence and the eleventh innovative applications of artificial intelligence conference innovative applications of artificial intelligence. American Association for Artificial Intelligence(AAAI/IAAI Orlando, Florida, USA), 844-849 2000.
18. Song Ruiping, A study on hybrid recommendation algorithm. Guanzhou: Lanzhou University, 2014
19. Tang M, Dai X, Cao B, et al. WSWalker: A Random Walk Method for QoS-Aware Web Service Recommendation. 2015 IEEE 22nd International Conference on Web Services. (ICWS New York, USA 2015). 591–598 (2015)
20. Zheng Z, Ma H, Lyu M R, et al, WSRec: A collaborative filtering based web service recommender system.Web Services, 2009. ICWS 2009. (IEEE International Conference on (Losangeles ,CA ,USA 2009). IEEE, 2009 p. 437-44
21. Social Computing Research at the University of Minnesota, MovieLens latest datasets [DB/OL]. http://www.grouplens.org/datasets/movielens/, 2016-01. 1 Mar 2016
22. NP Kumar, Z Fan, Hybrid user-item based collaborative filtering. Procedia. Comput. Sci. **60**(1), 1453–1461 (2015)
23. J Liu, B Li, W Zhang, Feature extraction using maximum variance sparse mapping. Neural. Comput. Appl. **21**(8), 1827–1833 (2012)
24. L* Jin, Z Jing et al., Irregular community discovery for social CRM in cloud computing. J. Supercomputing. **61**(2), 317–336 (2012)
25. X Luo, Z Xu, J Yu et al., Building association link network for semantic link on web resources. IEEE Trans. Automation. Sci. Eng. **8**(3), 482–494 (2011)
26. C Hu, Z Xu, Y Liu et al., Semantic link network-based model for organizing multimedia big data. IEEE Transactions. Emerg. Top. Comput. **2**(3), 376–387 (2014)
27. Z Xu, X Wei, X Luo et al., Knowle: a semantic link network based system for organizing large scale online news events. Future. Generation. Comput. Syst. **43**, 40–50 (2015)
28. Z Xu, X Luo, S Zhang et al., Mining temporal explicit and implicit semantic relations between entities using web search engines. Future. Generation. Comput. Syst. **37**, 468–477 (2014)
29. X Wei, X Luo, Q Li et al., Online comment-based hotel quality automatic assessment using improved fuzzy comprehensive evaluation and fuzzy cognitive map. IEEE Trans. Fuzzy Syst. **23**(1), 72–84 (2015)